BCSE309P

AES ALGORITHM

SANTHOSH KRISHNA R

21BCE1651

Client

```
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.net.Socket;
import java.nio.charset.StandardCharsets;
import java.util.Base64;

import javax.crypto.Cipher;
import javax.crypto.SecretKey;
import javax.crypto.SecretKeyFactory;
import javax.crypto.spec.IvParameterSpec;
import javax.crypto.spec.PBEKeySpec;
import javax.crypto.spec.SecretKeySpec;

public class Client {

   private static final String SECRET_KEY = "123456789";
   private static final String SALTVALUE = "abcdefg";

   public static void main(String[] args) {
      final String SERVER_ADDRESS = "localhost";
```

```
    final int SERVER_PORT = 12345;

    try {
        String originalMessage = "AES Encryption";

        // Encrypt message
        String encryptedMessage = encryptMessage(originalMessage);

        // Connect to server
        Socket socket = new Socket(SERVER_ADDRESS, SERVER_PORT);
        System.out.println("Connected to server.");

        DataInputStream dis = new DataInputStream(socket.getInputStream());
        DataOutputStream dos = new DataOutputStream(socket.getOutputStream());

        // Send encrypted message to server
        dos.writeInt(encryptedMessage.getBytes().length);
        dos.write(encryptedMessage.getBytes());
        dos.flush();

        // Close streams and socket
        dis.close();
        dos.close();
        socket.close();
    } catch (Exception e) {
        e.printStackTrace();
    }
}
```

```
    private static String encryptMessage(String strToEncrypt) {
        try {
            byte[] iv = new byte[16];
            IvParameterSpec ivspec = new IvParameterSpec(iv);
            SecretKeyFactory factory =
SecretKeyFactory.getInstance("PBKDF2WithHmacSHA256");
            KeySpec spec = new PBEKeySpec(SECRET_KEY.toCharArray(), SALTVALUE.getBytes(),
65536, 256);
            SecretKey tmp = factory.generateSecret(spec);
            SecretKeySpec secretKey = new SecretKeySpec(tmp.getEncoded(), "AES");
            Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5Padding");
            cipher.init(Cipher.ENCRYPT_MODE, secretKey, ivspec);
            byte[] encryptedBytes =
cipher.doFinal(strToEncrypt.getBytes(StandardCharsets.UTF_8));
            return Base64.getEncoder().encodeToString(encryptedBytes);
        } catch (Exception e) {
            e.printStackTrace();
            return null;
        }
    }
}
```

Server

```
import java.io.DataInputStream;
import java.io.DataOutputStream;
import java.net.ServerSocket;
import java.net.Socket;
import java.nio.charset.StandardCharsets;
import java.util.Base64;
```

```
import javax.crypto.Cipher;
import javax.crypto.SecretKey;
import javax.crypto.SecretKeyFactory;
import javax.crypto.spec.IvParameterSpec;
import javax.crypto.spec.PBEKeySpec;
import javax.crypto.spec.SecretKeySpec;

public class Server {

    private static final String SECRET_KEY = "123456789";
    private static final String SALTVALUE = "abcdefg";

    public static void main(String[] args) {
        final int SERVER_PORT = 12345;

        try {
            ServerSocket serverSocket = new ServerSocket(SERVER_PORT);
            System.out.println("Server started. Waiting for clients...");

            while (true) {
                Socket clientSocket = serverSocket.accept();
                System.out.println("Client connected: ");
                new Thread(() -> handleClient(clientSocket)).start();
            }
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
```

```
    private static void handleClient(Socket clientSocket) {
        try {
            DataInputStream dis = new DataInputStream(clientSocket.getInputStream());
            DataOutputStream dos = new DataOutputStream(clientSocket.getOutputStream());
            int length = dis.readInt();
            byte[] encryptedBytes = new byte[length];
            dis.readFully(encryptedBytes, 0, encryptedBytes.length);
            String decryptedMessage = decryptMessage(encryptedBytes);
            System.out.println("Decrypted message from client: " + decryptedMessage);
            dis.close();
            dos.close();
            clientSocket.close();
        } catch (Exception e) {
            e.printStackTrace();
        }
    }

    private static String decryptMessage(byte[] encryptedBytes) {
        try {
            byte[] iv = new byte[16];
            IvParameterSpec ivspec = new IvParameterSpec(iv);
            SecretKeyFactory factory =
SecretKeyFactory.getInstance("PBKDF2WithHmacSHA256");
            KeySpec spec = new PBEKeySpec(SECRET_KEY.toCharArray(), SALTVALUE.getBytes(),
65536, 256);
            SecretKey tmp = factory.generateSecret(spec);
            SecretKeySpec secretKey = new SecretKeySpec(tmp.getEncoded(), "AES");
            Cipher cipher = Cipher.getInstance("AES/CBC/PKCS5PADDING");
            cipher.init(Cipher.DECRYPT_MODE, secretKey, ivspec);
            byte[] decryptedBytes = cipher.doFinal(encryptedBytes);
```

```
            return new String(decryptedBytes, StandardCharsets.UTF_8);
        } catch (Exception e) {
            e.printStackTrace();
            return null;
        }
    }
}
```

Output: